

**This Page Is Inserted by IFW Operations
and is not a part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- **BLACK BORDERS**
- **TEXT CUT OFF AT TOP, BOTTOM OR SIDES**
- **FADED TEXT**
- **ILLEGIBLE TEXT**
- **SKEWED/SLANTED IMAGES**
- **COLORED PHOTOS**
- **BLACK OR VERY BLACK AND WHITE DARK PHOTOS**
- **GRAY SCALE DOCUMENTS**

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problems Mailbox.**

(19)



JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11) Publication number: **11296392 A**(43) Date of publication of application: **29.10.99**

(51) Int. Cl.

G06F 11/10
G06F 12/16
G06F 15/78

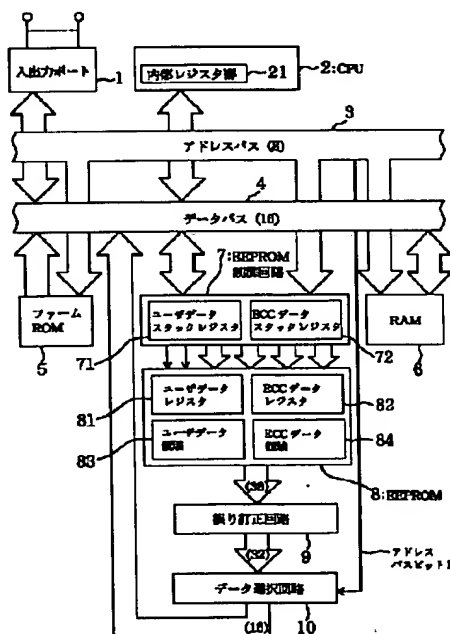
(21) Application number: **10097287**(71) Applicant: **NEC CORP**(22) Date of filing: **09.04.98**(72) Inventor: **FUKUSHIMA KIYOSHI**(54) **1 CHIP MICROCOMPUTER**

(57) Abstract:

PROBLEM TO BE SOLVED: To provide a 1 chip microcomputer for reducing the work load on a user at the time of preparing ECC(error collection code) data, preventing increase of an EEPROM(electrical erasable programmable read only memory) writing time based on the ECC data, and reducing a memory capacity.

SOLUTION: This 1 chip microcomputer is provided with a firm ROM 5 for storing a program for generating ECC data from user data and an EEPROM control circuit 7 for controlling wiring of user data and the ECC data in an EEPROM 8. A CPU 2 stores the ECC data generated by the program of the firm ROM 5 from the user data and the original user data in a user data area 83 and an ECC data area 84 of the EEPROM 8 according to the control of the EEPROM control circuit 7, and reads the user data in the user data area 83 and the ECC data in the ECC data area 84 for successively operating an error correction processing.

COPYRIGHT: (C)1999,JPO



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-296392

(43) 公開日 平成11年(1999)10月29日

(51) Int.Cl. ⁸	識別記号	F I
G 0 6 F 11/10	3 3 0	G 0 6 F 11/10
12/16	3 2 0	12/16
15/78	5 1 0	15/78
		3 3 0 K
		3 2 0 F
		5 1 0 K

審査請求 ☒ 有 請求項の数10 O L (全 20 頁)

(21) 出願番号 特願平10-97287

(22) 出願日 平成10年(1998)4月9日

(71) 出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(72) 発明者 福岡 清

東京都港区芝五丁目7番1号 日本電気株式会社社内

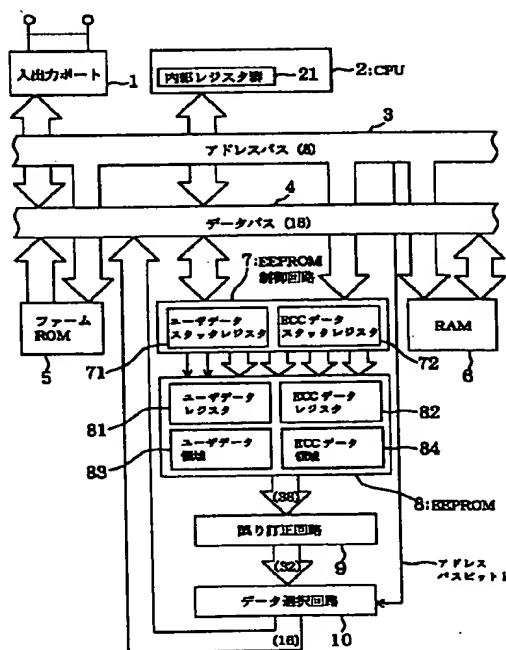
(74) 代理人 弁理士 西村 征生

(54) 【発明の名称】 1 チップマイクロコンピュータ

(57) 【要約】 (修正有)

【課題】 ECCデータ作成時におけるユーザの作業負担を軽減することができるとともに、ECCデータに基づくEEPROM書き込み時間の増加がなく、かつ、メモリ容量を削減できる1チップマイクロコンピュータを提供する。

【解決手段】 開示される1チップマイクロコンピュータは、ユーザデータからECCデータを生成するためのプログラムを格納したファームROM5と、EEPROM8に対するユーザデータとECCデータとの書き込みを制御するEEPROM制御回路7とを備え、CPU2がユーザデータからファームROM5のプログラムによって生成したECCデータとものユーザデータとを、EEPROM制御回路7の制御に応じてEEPROM8のユーザデータ領域83とECCデータ領域84とに格納し、ユーザデータ領域83のユーザデータとECCデータ領域84のECCデータとを読み出して誤り訂正の処理を順次行うように構成されている。



【特許請求の範囲】

【請求項 1】 ユーザデータと該ユーザデータに対応する ECC データとを格納する EEPROM を備え、該ユーザデータと ECC データとから誤り訂正を行ったユーザデータを発生するように構成されている 1 チップマイクロコンピュータにおいて、

ユーザデータから ECC データを生成するためのプログラムを格納した記憶手段と、前記 EEPROM に対するユーザデータと ECC データとの書き込みを制御する制御手段とを備え、CPU がユーザデータから該記憶手段のプログラムによって生成した ECC データとものとユーザデータとを、前記制御手段の制御に応じて前記 EEPROM のユーザデータ領域と ECC データ領域とに格納し、該ユーザデータ領域のユーザデータと ECC 領域の ECC データとを読み出して誤り訂正の処理を順次行うように構成されてなることを特徴とする 1 チップマイクロコンピュータ。

【請求項 2】 外部から前記 CPU の内部レジスタへ順次所定量のユーザデータを格納するステップと、前記 EEPROM 内のユーザデータレジスタへ前記所定量ずつユーザデータを転送するステップと、前記 CPU が格納されている所定量のユーザデータから ECC データを生成するステップと、前記 EEPROM 内の ECC データレジスタへ該 ECC データを転送するステップと、該ユーザデータレジスタのユーザデータと ECC データレジスタの ECC データとを、それぞれ前記 EEPROM の同じアドレスのユーザデータ領域と ECC データ領域とに書き込むステップとを順次実行し、該 EEPROM の同じアドレスのユーザデータと ECC データとを読み出して誤り訂正の処理を順次行うように構成されていることを特徴とする請求項 1 記載の 1 チップマイクロコンピュータ。

【請求項 3】 外部から前記制御手段内のユーザデータスタックレジスタに所定量のユーザデータを格納するステップと、外部から前記 CPU の内部レジスタへ所定量のユーザデータを格納するステップと、最初の処理サイクルにおいて、最初のアドレスのユーザデータを前記制御手段内のユーザデータレジスタに転送するステップと、該ユーザデータレジスタのユーザデータを前記 EEPROM 内のユーザデータ領域に格納するステップとを実行し、次の処理サイクル以降において、前記 CPU が前回の処理サイクルで格納されている所定量のユーザデータから ECC データを生成するステップと、前記制御手段内の ECC データスタックレジスタに該 ECC データを転送するステップと、前記 EEPROM 内のユーザデータレジスタにユーザデータを転送するとともに ECC データレジスタに該 ECC データを転送するステップと、EEPROM の ECC データ領域に対する書き込みアドレスを - 1 するステップと、前記ユーザデータレジスタのユーザデータと ECC データレジスタの ECC デ

ータとを、それぞれ前記 EEPROM のユーザデータ領域と ECC データ領域とに書き込むステップとを繰り返して実行し、

最後の処理サイクルにおいて、前記 CPU が前回の処理サイクルにおいて格納されている所定量のユーザデータから ECC データを生成するステップと、制御手段内の ECC データスタックレジスタに該 ECC データを転送するステップと、前記 EEPROM 内の ECC データレジスタに該 ECC データを転送するステップと、前記 EEPROM 内の ECC データ領域に対する書き込みアドレスを - 1 するステップと、該 ECC データレジスタの ECC データを前記 EEPROM の ECC データ領域に書き込むステップとを実行し、

該 EEPROM の同じアドレスのユーザデータと ECC データとを読み出して誤り訂正の処理を順次行うように構成されていることを特徴とする請求項 1 記載の 1 チップマイクロコンピュータ。

【請求項 4】 外部から前記制御手段内のユーザデータスタックレジスタに所定量のユーザデータを格納するステップと、外部から前記 CPU の内部レジスタへ所定量のユーザデータを格納するステップと、最初の処理サイクルにおいて、最初のアドレスのユーザデータを前記制御手段内のユーザデータレジスタに転送するステップと、該ユーザデータレジスタのユーザデータを前記 EEPROM 内のユーザデータ領域に格納するステップとを実行し、次の処理サイクル以降において、前記 CPU が前回の処理サイクルで格納されている所定量のユーザデータから ECC データを生成するステップと、前記制御手段内の ECC データスタックレジスタに該 ECC データを転送するステップと、前記 EEPROM 内のユーザデータレジスタにユーザデータを転送するとともに ECC データレジスタに該 ECC データを転送するステップと、前記ユーザデータレジスタのユーザデータと前記 EEPROM のユーザデータ領域の ECC データとを、それぞれ前記 EEPROM のユーザデータ領域と ECC データ領域とに書き込むステップとを繰り返して実行し、

最後の処理サイクルにおいて、前記 CPU が前回の処理サイクルにおいて格納されている所定量のユーザデータから ECC データを生成するステップと、制御手段内の ECC データスタックレジスタに該 ECC データを転送するステップと、前記 EEPROM 内の ECC データレジスタに該 ECC データを転送するステップと、該 ECC データレジスタの ECC データを前記 EEPROM の ECC データ領域に書き込むステップとを実行し、該 EEPROM のユーザデータと該ユーザデータのアドレスに + 1 したアドレスの ECC データとを読み出して誤り訂正の処理を順次行うように構成されていることを特徴とする請求項 1 記載の 1 チップマイクロコンピュータ。

【請求項 5】 外部から前記制御手段内のスタックレジスタへ順次所定量のユーザデータを転送するステップ

と、前記CPUの内部レジスタへ順次所定量のユーザデータを格納するステップと、該所定量のユーザデータをEEPROM内のユーザデータレジスタに転送するステップと、前記スタックレジスタのユーザデータを該スタックアドレスに対応するRAM領域に格納するステップとを実行するとともに、前記CPUが該RAM領域のデータからECCデータを生成するステップと、該ECCデータをEEPROM内のECCデータレジスタに転送するステップとを繰り返して実行し、

前記制御手段から前記EEPROMのユーザデータ領域のアドレスとECCデータ領域のアドレスとを出力するステップと、前記ユーザデータレジスタのユーザデータとECCデータレジスタのECCデータとをEEPROMのユーザデータ領域とECCデータ領域のそれぞれの指定アドレスに格納するステップとを繰り返して実行し、該EEPROMの同じアドレスのユーザデータとECCデータとを読み出して誤り訂正の処理を順次行うように構成されていることを特徴とする請求項1記載の1チップマイクロコンピュータ。

【請求項6】 ユーザデータと該ユーザデータに対応するECCデータとを格納するEEPROMを備え、該ユーザデータとECCデータとから誤り訂正を行ったユーザデータを発生するように構成されている1チップマイクロコンピュータにおいて、ユーザデータからECCデータを生成するためのプログラムを内蔵し、該プログラムを用いて入力ユーザデータから対応するECCデータを生成する処理を装置内部において行うようにしたことを特徴とする1チップマイクロコンピュータ。

【請求項7】 外部から順次所定量のユーザデータを入力して、前記CPUが順次該所定量のユーザデータからECCデータを生成し、前記ユーザデータと対応するECCデータとを、それぞれ前記EEPROMの同じアドレスのユーザデータ領域とECCデータ領域とに書き込むとともに、該EEPROMの同じアドレスのユーザデータとECCデータとを読み出して誤り訂正の処理を順次行うように構成されていることを特徴とする請求項6記載の1チップマイクロコンピュータ。

【請求項8】 外部から順次所定量のユーザデータを入力して、前記CPUが順次該所定量のユーザデータから1処理サイクル後までにECCデータを生成し、前記ユーザデータと対応する1処理サイクル後のECCデータとを、それぞれ前記EEPROMのユーザデータ領域と-1したアドレスのECCデータ領域とに書き込むとともに、該EEPROMの同じアドレスのユーザデータとECCデータとを読み出して誤り訂正の処理を順次行うように構成されていることを特徴とする請求項6記載の1チップマイクロコンピュータ。

【請求項9】 外部から順次所定量のユーザデータを入力して、前記CPUが順次該所定量のユーザデータから1処理サイクル後までにECCデータを生成し、前記ユ

ーザデータと対応する1処理サイクル後のECCデータとを、それぞれ前記EEPROMの同じアドレスのユーザデータ領域とECCデータ領域とに書き込むとともに、該EEPROMのユーザデータと+1したアドレスのECCデータとを読み出して誤り訂正の処理を順次行うように構成されていることを特徴とする請求項6記載の1チップマイクロコンピュータ。

【請求項10】 外部から順次所定量のユーザデータを入力して、前記CPUが逐次該所定量のユーザデータからECCデータを生成するとともに、次のユーザデータの入力時までECCデータの生成を終了しないときは無効データを出力し、前記ユーザデータと対応する有効なECCデータとを、それぞれ前記EEPROMの同じアドレスのユーザデータ領域とECCデータ領域とに書き込むとともに、該EEPROMの同じアドレスのユーザデータとECCデータとを読み出して誤り訂正の処理を順次行うように構成されていることを特徴とする請求項6記載の1チップマイクロコンピュータ。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 この発明は、ECC (Error Correction Code) データを内部的に生成、付加することが可能な1チップマイクロコンピュータに関する。

【0002】

【従来の技術】 EEPROM (Electrically Erasable Programmable Read Only Memory) はデータの消去、書き換えが可能なメモリ (電氣的に一括して消去可能なフラッシュEEPROMを含む) であって、1チップマイクロコンピュータ等において、プログラムの格納用等として多く用いられている。EEPROMにおける保持データの信頼性を向上するためには、ユーザデータにECCデータを付加して記録し、読み出されたユーザデータについてECCデータを用いて誤り訂正を行うことが有効である。従来、このようなECCデータの付加は、ユーザデータから専用のソフトウェアを使用してECCデータを生成して、ユーザデータとともにEEPROMに格納することによって行われていた。例えば、自動車のエンジン制御などに使用される1チップマイクロコンピュータは、制御プログラムによってエンジンの回転数や燃料噴射などの重要な制御を行っている。この制御プログラムを記憶しているROM (Read Only Memory) の内容がなんらかの理由によって変化した場合、1チップマイクロコンピュータは異常な処理を行うことになるので、危険な状態を招きかねない。そのため、ユーザがROMにユーザデータ (プログラムなど) を記憶させるとき、ECCデータを付加して記憶させておいて、1チップマイクロコンピュータがユーザデータを読み出したとき1ビットの誤りであれば訂正して処理し、2ビット以上の誤りであれば異常表示を行うなどの処理を行うようにしている。

【0003】さらに、ROMに記憶させたユーザデータ（プログラム）にバグが発見されたとき、自動車メーカーは、販売済みの自動車を回収して1チップマイクロコンピュータを交換する作業が必要になる。この場合、ROMをマスクROMなどのような書き換え不可能なROMで構成していると、自動車メーカーは1チップマイクロコンピュータを含むエンジン制御ユニットごと交換しなければならないが、そうすると、ユニット代だけでなく、取り替えコストもかかるので、莫大な損失を受けることになる。そこで、1チップマイクロコンピュータのROMをEEPROMで構成しておけば、自動車メーカーはエンジン制御ユニットのコネクタを介してプログラムを書き換えるだけで済むので、バグ対応費用を大幅に低減できる。

【0004】図20は、EEPROMにおける従来のECCデータの生成・付加方法の説明図である。以下、図20を参照して、従来技術を説明する。まず、ユーザはECCデータ作成のために、所定のプログラムを作成する（同図（a）101）。いま、ユーザデータ102として、00000H～0FFFFHまでのデータを入力したものとする。ユーザは、作成されたプログラムによって、ユーザデータ102に基づいて専用のソフトウェアを用いてECCデータを生成する（同図（a）103）。これによって、ECCデータ104として、10000H～14FFFHまでのデータが生成される。次に、ユーザデータ102にECCデータ104を付加して、書き込みデータ105として00000H～14FFFHまでのデータを生成し、図示されないマイクロコンピュータに内蔵するEEPROMに対する書き込み（同図（a）106）を行う。この際におけるデータの書き込みは、専用のライターまたはオンボード書き込みによって実行される。同図20（b）は、EEPROM上のアドレスマップを示したものであって、ユーザデータとして、00000H～0FFFFHが割り当てられ、ECCデータとして、10000H～14FFFHが割り当てられている。ECCデータのボリュームは、16ビットデータの場合5ビット必要であり、このため、同図（b）に示すようなデータ領域が必要になる。

【0005】

【発明が解決しようとする課題】しかしながら、上記従来のECCデータの生成・付加方法にあっては、ユーザデータの処理とは別にECCデータを生成するようにしていたため、ユーザデータを基にECCデータを生成する専用ソフトウェアが必要である、という問題があった。また、ECCデータは、ユーザデータの後のアドレスにマッピングされるため、00000H～0FFFFHのユーザデータに加えて、ECCデータとして10000H～14FFFHのデータを書き込むので、このため、書き込み時間が約30パーセント増加するという問題があった。さらに、ECCデータによって書き込みデ

ータが全体として約30パーセント増加するため、システム全体の外部メモリ容量を約30パーセント増加しなければならないという問題があった。

【0006】また、論理回路で構成したECCデータの生成回路を、1チップマイクロコンピュータに内蔵した構成も既に知られている。近年において、1チップマイクロコンピュータのデータバスの幅は増加する傾向にあり、従来、8ビット幅であったものが、32ビット幅や64ビット幅に変わりつつある。そこでECCデータの生成回路を論理回路で構成すると、データバス幅が増加するのに伴って、回路規模は指数関数的に増加し、半導体チップに占める面積が増加する。EEPROMを1チップマイクロコンピュータのプログラム格納用に用いる場合、ECCデータの生成回路は、プログラムを格納する際のみ使用するものであって、プログラム実行時には使用されない回路である。それにもかかわらず、回路規模の大きいECCデータの生成回路を内蔵することは、1チップマイクロコンピュータのコストパフォーマンスを悪化させる原因になる。

【0007】この発明は、上述の事情に鑑みてなされたものであって、ECCデータ作成時におけるユーザの作業負担を軽減することができるとともに、ECCデータに基づくEEPROM書き込み時間の増加がなく、かつ、メモリ容量を削減できる1チップマイクロコンピュータを提供することを目的としている。

【0008】

【課題を解決するための手段】上記課題を解決するために、請求項1記載の発明は、ユーザデータと該ユーザデータに対応するECCデータとを格納するEEPROMを備え、該ユーザデータとECCデータとから誤り訂正を行ったユーザデータを発生するように構成されている1チップマイクロコンピュータに係り、ユーザデータからECCデータを生成するためのプログラムを格納した記憶手段と、上記EEPROMに対するユーザデータとECCデータとの書き込みを制御する制御手段とを備え、CPUがユーザデータから該記憶手段のプログラムによって生成したECCデータともとのユーザデータとを、上記制御手段の制御に応じて上記EEPROMのユーザデータ領域とECCデータ領域とに格納し、該ユーザデータ領域のユーザデータとECC領域のECCデータとを読み出して誤り訂正の処理を順次行うように構成されてなることを特徴としている。

【0009】また、請求項2記載の発明は、請求項1記載の1チップマイクロコンピュータに係り、外部から上記CPUの内部レジスタへ順次所定量のユーザデータを格納するステップと、上記EEPROM内のユーザデータレジスタへ上記所定量ずつユーザデータを転送するステップと、上記CPUが格納されている所定量のユーザデータからECCデータを生成するステップと、上記EEPROM内のECCデータレジスタへ該ECCデータ

を転送するステップと、該ユーザデータレジスタのユーザデータとECCデータレジスタのECCデータとを、それぞれ上記EEPROMの同じアドレスのユーザデータ領域とECCデータ領域とに書き込むステップとを順次実行し、該EEPROMの同じアドレスのユーザデータとECCデータとを読み出して誤り訂正の処理を順次行うように構成されていることを特徴としている。

【0010】また、請求項3記載の発明は、請求項1記載の1チップマイクロコンピュータに係り、外部から上記制御手段内のユーザデータスタックレジスタに所定量のユーザデータを格納するステップと、外部から上記CPUの内部レジスタへ所定量のユーザデータを格納するステップと、最初の処理サイクルにおいて、最初のアドレスのユーザデータを上記制御手段内のユーザデータレジスタに転送するステップと、該ユーザデータレジスタのユーザデータを上記EEPROM内のユーザデータ領域に格納するステップとを実行し、次の処理サイクル以降において、上記CPUが前回の処理サイクルで格納されている所定量のユーザデータからECCデータを生成するステップと、上記制御手段内のECCデータスタックレジスタに該ECCデータを転送するステップと、上記EEPROM内のユーザデータレジスタにユーザデータを転送するとともにECCデータレジスタに該ECCデータを転送するステップと、EEPROMのECCデータ領域に対する書き込みアドレスを-1するステップと、上記ユーザデータレジスタのユーザデータとECCデータレジスタのECCデータとを、それぞれ上記EEPROMのユーザデータ領域とECCデータ領域とに書き込むステップとを繰り返して実行し、最後の処理サイクルにおいて、上記CPUが前回の処理サイクルにおいて格納されている所定量のユーザデータからECCデータを生成するステップと、制御手段内のECCデータスタックレジスタに該ECCデータを転送するステップと、上記EEPROM内のECCデータレジスタに該ECCデータを転送するステップと、上記EEPROM内のECCデータ領域に対する書き込みアドレスを-1するステップと、該ECCデータレジスタのECCデータを上記EEPROMのECCデータ領域に書き込むステップとを実行し、該EEPROMの同じアドレスのユーザデータとECCデータとを読み出して誤り訂正の処理を順次行うように構成されていることを特徴としている。

【0011】また、請求項4記載の発明は、請求項1記載の1チップマイクロコンピュータに係り、外部から上記制御手段内のユーザデータスタックレジスタに所定量のユーザデータを格納するステップと、外部から上記CPUの内部レジスタへ所定量のユーザデータを格納するステップと、最初の処理サイクルにおいて、最初のアドレスのユーザデータを上記制御手段内のユーザデータレジスタに転送するステップと、該ユーザデータレジスタ

のユーザデータを上記EEPROM内のユーザデータ領域に格納するステップとを実行し、次の処理サイクル以降において、上記CPUが前回の処理サイクルで格納されている所定量のユーザデータからECCデータを生成するステップと、上記制御手段内のECCデータスタックレジスタに該ECCデータを転送するステップと、上記EEPROM内のユーザデータレジスタにユーザデータを転送するとともにECCデータレジスタに該ECCデータを転送するステップと、上記ユーザデータレジスタのユーザデータとECCデータレジスタのECCデータとを、それぞれ上記EEPROMのユーザデータ領域とECCデータ領域とに書き込むステップとを繰り返して実行し、最後の処理サイクルにおいて、上記CPUが前回の処理サイクルにおいて格納されている所定量のユーザデータからECCデータを生成するステップと、制御手段内のECCデータスタックレジスタに該ECCデータを転送するステップと、上記EEPROM内のECCデータレジスタに該ECCデータを転送するステップと、該ECCデータレジスタのECCデータを上記EEPROMのECCデータ領域に書き込むステップとを実行し、該EEPROMのユーザデータと該ユーザデータのアドレスに+1したアドレスのECCデータとを読み出して誤り訂正の処理を順次行うように構成されていることを特徴としている。

【0012】また、請求項5記載の発明は、請求項1記載の1チップマイクロコンピュータに係り、外部から上記制御手段内のスタックレジスタへ順次所定量のユーザデータを転送するステップと、上記CPUの内部レジスタへ順次所定量のユーザデータを格納するステップと、該所定量のユーザデータをEEPROM内のユーザデータレジスタに転送するステップと、上記スタックレジスタのユーザデータを該スタックアドレスに対応するRAM領域に格納するステップとを実行するとともに、上記CPUが該RAM領域のデータからECCデータを生成するステップと、該ECCデータをEEPROM内のECCデータレジスタに転送するステップとを繰り返して実行し、上記制御手段から上記EEPROMのユーザデータ領域のアドレスとECCデータ領域のアドレスとを出力するステップと、上記ユーザデータレジスタのユーザデータとECCデータレジスタのECCデータとをEEPROMのユーザデータ領域とECCデータ領域のそれぞれの指定アドレスに格納するステップとを繰り返して実行し、該EEPROMの同じアドレスのユーザデータとECCデータとを読み出して誤り訂正の処理を順次行うように構成されていることを特徴としている。

【0013】さらにまた、請求項6記載の発明は、ユーザデータと該ユーザデータに対応するECCデータとを格納するEEPROMを備え、該ユーザデータとECCデータとから誤り訂正を行ったユーザデータを発生するように構成されている1チップマイクロコンピュータに

係り、ユーザデータからECCデータを生成するためのプログラムを内蔵し、該プログラムを用いて入力ユーザデータから対応するECCデータを生成する処理を装置内部において行うようにしたことを特徴としている。

【0014】また、請求項7記載の発明は、請求項6記載の発明に係る1チップマイクロコンピュータであって、外部から順次所定量のユーザデータを入力して、上記CPUが順次該所定量のユーザデータからECCデータを生成し、上記ユーザデータと対応するECCデータとを、それぞれ上記EEPROMの同じアドレスのユーザデータ領域とECCデータ領域とに書き込むとともに、該EEPROMの同じアドレスのユーザデータとECCデータとを読み出して誤り訂正の処理を順次行うように構成されていることを特徴としている。

【0015】また、請求項8記載の発明は、請求項6記載の発明に係る1チップマイクロコンピュータであって、外部から順次所定量のユーザデータを入力して、上記CPUが順次該所定量のユーザデータから1処理サイクル後までにECCデータを生成し、上記ユーザデータと対応する1処理サイクル後のECCデータとを、それぞれ上記EEPROMのユーザデータ領域と-1したアドレスのECCデータ領域とに書き込むとともに、該EEPROMの同じアドレスのユーザデータとECCデータとを読み出して誤り訂正の処理を順次行うように構成されていることを特徴としている。

【0016】また、請求項9記載の発明は、請求項6記載の発明に係る1チップマイクロコンピュータであって、外部から順次所定量のユーザデータを入力して、上記CPUが順次該所定量のユーザデータから1処理サイクル後までにECCデータを生成し、上記ユーザデータと対応する1処理サイクル後のECCデータとを、それぞれ上記EEPROMの同じアドレスのユーザデータ領域とECCデータ領域とに書き込むとともに、該EEPROMのユーザデータと+1したアドレスのECCデータとを読み出して誤り訂正の処理を順次行うように構成されていることを特徴としている。

【0017】また、請求項10記載の発明は、請求項6記載の発明に係る1チップマイクロコンピュータであって、外部から順次所定量のユーザデータを入力して、上記CPUが逐次該所定量のユーザデータからECCデータを生成するとともに、次のユーザデータの入力時までにECCデータの生成を終了しないときは無効データを出力し、上記ユーザデータと対応する有効なECCデータとを、それぞれ上記EEPROMの同じアドレスのユーザデータ領域とECCデータ領域とに書き込むとともに、該EEPROMの同じアドレスのユーザデータとECCデータとを読み出して誤り訂正の処理を順次行うように構成されていることを特徴としている。

【0018】

【作用】この発明の構成では、外部からユーザデータを

入力して、CPUが内部プログラムに基づいて順次、所定量のユーザデータからECCデータを生成して、ユーザデータと対応するECCデータとを、それぞれEEPROMのユーザデータ領域とECCデータ領域とに格納するようにしたので、外部的にECCデータを生成して付加する場合と比べて、ユーザの負担が大幅に減少するとともに、ECCデータ作成のために、作成したECCデータを一旦格納する外部メモリを必要としないので、システムの装置のコストを低減することができ、さらに、EEPROMに対して、ユーザデータとECCデータとを同時に書き込むので、ECCデータを外部メモリから書き込む場合と比べて、EEPROMに対するデータ書き込み時間を低減することができる。

【0019】

【発明の実施の形態】以下、図面を参照して、この発明の実施の形態について説明する。説明は、実施例を用いて具体的にこなう。図1は、この発明の一実施例である1チップマイクロコンピュータの電氣的構成を示す図、図2は、EEPROM制御回路の構成例を示す図、図3は、データラッチ選択制御回路の構成例を示す図、図4は、書き込み信号生成回路の構成例を示す図、図5は、書き込み信号生成回路の動作タイミングを示す図、図6、図7は、この例の第1の動作例を示すフローチャート、図8は、第1の動作例における書き込み時のタイミングチャート、図9、図10は、この例の第2の動作例を示すフローチャート、図11は、書き込み時ECCアドレスをシフトする場合のタイミングチャート、図12は、書き込み時ECCアドレスをシフトする場合のEEPROMアドレス生成回路の構成例を示す図、図13、図14は、この例の第3の動作例を示すフローチャート、図15は、読み出し時ECCアドレスをシフトする場合のタイミングチャート、図16は、読み出し時ECCアドレスをシフトする場合のEEPROMアドレス生成回路の構成例を示す図、図17、図18は、この例の第4の動作例を示すフローチャート、図19は、第4の動作例における書き込み時のタイミングチャートである。この例の1チップマイクロコンピュータは、図1に示すように、入出力ポート1と、CPU(Central Processing Unit)2と、アドレスバス3と、データバス4と、ファームROM5と、RAM(Random Access Memory)6と、EEPROM制御回路7と、EEPROM8と、誤り訂正回路9と、データ選択回路10とから概略構成されている。

【0020】入出力ポート1は、外部との間でデータの入出力を行う複数の端末ポートからなっている。CPU2は、プログラムによって与えられる命令の解釈と実行を制御することによって、この例の1チップマイクロコンピュータの全体の動作を制御する。アドレスバス3は、この例の1チップマイクロコンピュータの各部の間で、アドレス信号を転送する。データバス4は16ビット

ト幅からなり、この例の1チップマイクロコンピュータの各部の間で、データ信号を転送する。ファームROM 5は、1チップマイクロコンピュータの動作に必要なファームウェア（マイクロプログラム）を格納するものであって、この例の場合は特にECCデータ生成用プログラムを収容している。RAM 6は、主としてCPU 2の作業領域として用いられる随時読み出し書き込みメモリである。

【0021】EEPROM制御回路7は、EEPROM 8の書き込み、読み出しを制御するものであって、後述する内容を有するとともに、データスタックのために、ユーザデータスタックレジスタ71とECCデータスタックレジスタ72とを有している。EEPROM 8は、電気的手段によって情報の書き込み、消去が可能な固定記憶素子であって、データ書き込みのために、ユーザデータレジスタ81とECCデータレジスタ82とを有するとともに、データ格納のために、ユーザデータ領域83とECCデータ領域84とを有している。なお、EEPROM 8は、通常のEEPROMとフラッシュEEPROMのいずれでもよい。誤り訂正回路9は、データに付加されているECCデータによって、データの誤りを検出して訂正する。データ選択回路10は、32ビットからなる誤り訂正回路9の出力を上位16ビットと下位16ビットとに分割して、交互にデータバス4に転送する。

【0022】また、上記EEPROM制御回路7は、図2に示すように、アドレスバス11と、データバス12と、データラッチ選択制御回路13と、アドレスnのデータラッチ14と、アドレスn+2のデータラッチ15と、ECCのデータラッチ16と、書き込み信号生成回路17と、EEPROMアドレス生成回路18とを含んで構成されている。ここでは、外部からのユーザデータのアドレスnを基準に説明する。アドレスバス11は、アドレスバス3の一部であって、アドレス信号を転送する。データバス12は、データバス4の一部であって、データ信号を転送する。データラッチ選択制御回路13は、データバス12上のデータを、アドレスnのデータラッチ14、アドレスn+2のデータラッチ15、ECCのデータラッチ16において選択的にラッチするための制御信号DL1、DL2、DL3を、アドレスバス11の情報に従って発生するものであって、その構成は後述する。アドレスnのデータラッチ14は、制御信号DL1がアクティブのとき、データバス12上のアドレスnとn+1の2バイト（16ビット）のデータをラッチするものであって、アドレスのビット1が“0”のとき、データをラッチしてEEPROM 8のメモリセルへ出力する。

【0023】アドレスn+2のデータラッチ15は、制御信号DL2がアクティブのとき、データバス12上のアドレスn+2とn+3の2バイト（16ビット）のデ

ータをラッチするものであって、アドレスのビット1が“1”のとき、データをラッチしてEEPROM 8のメモリセルへ出力する。ECCのデータラッチ16は、制御信号DL3がアクティブのとき、データバス12上の下位6ビットのデータをラッチするものであって、ECCアドレスが出力されたとき、データをラッチしてEEPROM 8のメモリセルへ出力する。

【0024】書き込み信号生成回路17は、アドレスバス11の情報とデータバス12の情報とから、EEPROM 8に対する書き込み開始を制御するための書き込みセット信号と、書き込み終了を制御するための書き込みリセット信号とを発生するものであって、その構成は後述する。EEPROMアドレス生成回路18は、アドレスバス11の情報からEEPROM 8に対するアドレスを生成するものであって、アドレスバス11のビット2～15を、EEPROM 8のデータ領域（ユーザメモリ）用のアドレスデコーダ（不図示）を介して、EEPROM 8のデータ領域のアドレス0～13として与え、アドレスバス11のビット2～15をディクリメント回路（不図示）とEEPROM 8のECCデータ領域のアドレスデコーダ（不図示）を介して、EEPROM 8のECCデータ領域のアドレス0～13として与える。なお、このようなアドレスの変換を行うのは、アドレス信号がバイト単位（8ビット）であるのに対して、EEPROM 8の32ビットの入出力データに対応する対応するアドレスとしては、最下位の2ビットが不要なためである。なお、アドレスnのデータラッチ14とアドレスn+2のデータラッチ15は、データが16ビット幅なので、外部のアドレスnに対しては、それぞれn、n+1と、n+2、n+3のアドレスのデータをラッチする。

【0025】データラッチ選択制御回路13は、図3に示すように、アドレスバス131と、ノア回路132と、インバータ133と、ノア回路134とを含んで構成されている。アドレスバス131は、アドレスバス11の一部であって、アドレス信号を転送する。ノア回路132は、EEPROM 8のデータ領域を表すアドレスバスのビット1と、ECCデータ領域を示すアドレスバス131のビット16とがインアクティブのとき、アドレスnのデータラッチ14に対する制御信号DL1を出力する。インバータ133とノア回路134は、EEPROM 8のデータ領域を表すアドレスバスのビット1がアクティブで、ECCデータ領域を示すアドレスバス131のビット16とがインアクティブのとき、アドレスn+2のデータラッチ15に対する制御信号DL2を出力する。一方、アドレスバス131のビット16がアクティブのときは、ECCのデータラッチ16に対する制御信号DL3を出力する。

【0026】書き込み信号生成回路17は、図4に示すように、アドレスバス171と、データバス172と、

アドレスデコーダ173と、FF (Flip Flop) 174と、FF175と、インバータ176と、アンド回路177とを含んで構成されている。書き込み信号生成回路17の動作は、図5のタイミングチャートによって示される。アドレスバス171は、アドレスバス11の一部であって、アドレス信号を転送する。データバス172はデータバス12の一部であって、データ信号を転送する。アドレスデコーダ173は、CPU2の出力したアドレスバス171上のアドレスが、EEPROM8に対する書き込み信号の生成を指示する、予め設定されている特定のアドレスと一致したことを検出したとき、アドレス一致信号を出力する。FF174とFF175は、アドレス一致信号が出力されたとき、それぞれデータバス172のデータビット0とデータビット1をラッチする。これによって、データビット1に応じて書き込みリセット信号が出力され、データビット0が出力されたとき、書き込みリセット信号が出力されていないことを条件として、アンド回路177から書き込みセット信号が出力される。EEPROM8では、書き込みセット信号によって書き込みパルスを立ち上げ、書き込みリセット信号によって書き込みパルスを立ち下げることによって、書き込みパルスがアクティブの期間にユーザデータとECCデータの書き込みを行う。

【0027】次に、図6、図7、図8を参照して、この例の第1の動作例を説明する。この第1の動作例は、EEPROM8の書き込みが遅いため、書き込みパルス幅内でECCデータを生成できる場合の処理方法を示している。ユーザデータを収容した外部のファイル（不図示）における、ユーザデータの最初のアドレスと最後のアドレスとが予め指定されているものとする。最初のアドレスから入出力ポート1へユーザデータを入力し（ステップS1）、データバス4を介してCPU2の内部レジスタ群21へユーザデータを格納する（ステップS2）処理を繰り返すことによって、4バイトデータを格納した（ステップS3）後、内部レジスタ群21からデータバス4を介してEEPROM制御回路7内のユーザデータスタックレジスタ71へユーザデータを転送し（ステップS4）、さらにEEPROM8内のユーザデータレジスタ81にユーザデータを転送する（ステップS5）。

【0028】一方、CPU2はファームROM5に格納したECC生成プログラムを読み出して実行することによって、格納した4バイトデータによってECCデータを生成し（ステップS6）、データバス4を介してEEPROM制御回路7内のECCデータスタックレジスタ72へECCデータを転送し（ステップS7）、さらにEEPROM8内のECCデータレジスタ82へECCデータを転送する（ステップS8）。そして、EEPROM制御回路7からEEPROM8へ書き込みセット信号を与える（ステップS9）ことによって、ユーザデー

タレジスタ81のユーザデータとECCデータレジスタ82のECCデータとを、それぞれEEPROM8のユーザデータ領域83とECCデータ領域84に書き込む処理を、4バイトデータのすべてのデータ書き込みが終了するまで繰り返して行い（ステップS10）、書き込み終了時、EEPROM8へ書き込みリセット信号を与える（ステップS11）。4バイトデータの処理を終了したとき、ユーザデータのアドレスを+4して（ステップS13）、再びステップS5、S8から処理を繰り返す、最後のアドレスになった（ステップS12）とき処理を終了する。

【0029】この場合の各データの書き込みタイミングは、図8に例示されるようになる。すなわち、最初、ポート1から4バイトのユーザデータを書き込みデータ（データ0～データ3）として入力することによって、このデータがデータバス4に連続して出力されている。これを所定のタイミングでEEPROM8内のユーザデータレジスタ81に転送し、さらにCPU2においてデータ0～データ3からのECCデータの生成が終了して、EEPROM8内のECCデータレジスタ82に転送されたとき、書き込みパルスを発生して、EEPROM8に書き込む。EEPROM8のデータを読み出すときは、同一アドレスからユーザデータとECCデータとを読み出すことによって、対応するユーザデータ部分とECCデータ部分とを読み出すことができるので、誤り訂正回路9ではこれによって誤り訂正の処理を行うことができる。

【0030】次に、図9、図10、図11及び図12を参照して、この例の第2の動作例について説明する。この第2の動作例は、EEPROM8の書き込みが遅いため、書き込みパルス幅内でECCデータを生成できない場合であって、書き込み時ECCアドレスをシフトする場合の処理方法を示している。ユーザデータを収容した外部のファイル（不図示）における、ユーザデータの最初のアドレスと最後のアドレスとが予め指定されている。最初のアドレスから入出力ポート1へユーザデータを入力し（ステップP1）、データバス4を介してEEPROM制御回路7内のユーザデータスタックレジスタ71へユーザデータを転送し（ステップP2）、さらに入力されたデータをデータバス4を介してCPU2の内部レジスタ群21へ格納する（ステップP3）処理を繰り返すことによって、4バイトデータを格納した（ステップP4）後、この4バイトデータが最初のアドレスに対応するものであった（ステップP5）ときは、ユーザデータスタックレジスタ71からEEPROM8内のユーザデータレジスタ81へユーザデータを転送し（ステップP6）、EEPROM制御回路7からEEPROM8へ書き込みセット信号を出力して（ステップP7）、EEPROM8に対する書き込みを行って、書き込みが終了した（ステップP8）とき、EEPROM制御回路

7からEEPROM8へ書き込みリセット信号を出力し（ステップP9）、次にアドレスを+4して（ステップP10）、再びステップP1に戻って、外部からの4バイトデータの入力から繰り返して実行する。

【0031】4バイトデータの格納終了時、アドレスが最初のアドレスに対応するものでなく（ステップP5）、最終のアドレスに対応するものでない（ステップP11）ときは、CPU2が前回の処理サイクルで格納されている4バイトデータからECCデータを生成して（ステップP12）、データバス4を介してEEPROM制御回路7内のECCデータスタックレジスタ72へECCデータを転送し（ステップP13）、EEPROM8内のユーザデータレジスタ81へユーザデータスタックレジスタ71からユーザデータを転送し、ECCデータレジスタ82へECCデータスタックレジスタ72からECCデータを転送し（ステップP14）、EEPROM8のECCデータ領域のアドレスを-1シフトし（ステップP15）、EEPROM8への書き込みセット信号を出力して（ステップP16）、EEPROM8のユーザデータ領域83にユーザデータレジスタ81からユーザデータを書き込み、ECCデータ領域84にECCデータレジスタ82からECCデータの書き込みを行う。そして書き込みが終了した（ステップP17）とき、EEPROM8への書き込みリセット信号を出力し（P18）、ステップP10に戻って、アドレスを+4シフトして（ステップP10）、再びステップP1に戻って、外部からの4バイトデータの入力から繰り返して実行する。ステップP11において、最後のアドレスになったときは、CPU2が格納した4バイトデータからECCデータを生成し（ステップP19）、データバス4を介してEEPROM制御回路7内のECCデータスタックレジスタ72へECCデータを転送し（ステップP20）、EEPROM8内のECCデータレジスタ82へECCデータを転送し（ステップP21）、EEPROM8のECCデータ領域84のアドレスを-1シフトし（ステップP22）、EEPROM8への書き込みセット信号を出力して（ステップP23）ECCデータの書き込みを行う。そして書き込みが終了した（ステップP24）とき、処理を終了する。

【0032】書き込み時ECCアドレスをシフトする場合の各データの書き込みタイミングは、図11に例示されるようになる。すなわち、ポート1から4バイトからなるページ単位に、ユーザデータを、ページアドレス0, 1, 2, ..., n, ..., 最終アドレスの順にEEPROM8内のユーザデータレジスタ81に転送するとともに、これに基づいて1サイクル遅れて生成された対応するECCデータを、順次1アドレスずつシフトしてEEPROM8のECCデータレジスタ82に転送し、書き込みパルスに応じて、それぞれEEPROM8のユーザデータ領域83とECCデータ領域84に書き込む。誤

り訂正処理のためにEEPROM8のデータを読み出すときは、EEPROM8の同一アドレスから、ユーザデータと、ECCデータとを読み出すことによって、対応するユーザデータ部分とECCデータ部分とを同時に読み出すことができる。

【0033】書き込み時ECCアドレスをシフトする場合の、EEPROM制御回路7における、EEPROMアドレス生成回路18は、図12に示すように、アドレスバス181と、ディクリメント回路182とを含んで構成されている。アドレスバス181は、アドレスバス11の一部であって、アドレス信号を転送する。ディクリメント回路182は、アドレスバス181のアドレスから1減算して出力する。EEPROM8へは、32ビットのユーザデータと、6ビットのECCデータとを同時に書き込むので、アドレスバス181のビット2~15を、EEPROM8のユーザデータ領域83の書き込みアドレスとして、EEPROM8のアドレス0~13に与え、アドレスバス181のビット2~15を、ディクリメント回路182を介して1を減算して、EEPROM8のECCデータ領域84の書き込み用として、EEPROM8のアドレス0~13に与える。このような変換を行うのは、アドレス信号がバイト単位（8ビット）であるのに対して、EEPROM8の32ビットの入出力データに対応する対応するアドレスとしては、最下位の2ビットが不要なためである。

【0034】次に、図13、図14、図15及び図16を参照して、この例の第3の動作例について説明する。この第3の動作例は、EEPROM8の書き込みが速いため、書き込みパルス幅内でECCデータを生成できない場合であって、読み出し時ECCアドレスをシフトする場合の処理方法を示している。ユーザデータを収容した外部の20イル（不図示）における、ユーザデータの最初のアドレスと最後のアドレスとが予め指定されている。最初のアドレスから入出力ポート1へユーザデータを入力し（ステップR1）、データバス4を介してEEPROM制御回路7内のユーザデータスタックレジスタ71へユーザデータを転送し（ステップR2）、さらに入力されたデータをデータバス4を介してCPU2の内部レジスタ群21へ格納する（ステップR3）処理を繰り返すことによって、4バイトデータを格納した（ステップR4）後、この4バイトデータが最初のアドレスに対応するものであった（ステップR5）ときは、ユーザデータスタックレジスタ71からEEPROM8内のユーザデータレジスタ81へユーザデータを転送し（ステップR6）、EEPROM制御回路7からEEPROM8へ書き込みセット信号を出力して（ステップR7）、EEPROM8に対する書き込みを行って、書き込みが終了した（ステップR8）とき、EEPROM制御回路7からEEPROM8へ書き込みリセット信号を出力し（ステップR9）、次にアドレスを+4して（ステップ

R10)、再びステップR1に戻って、外部からの4バイトデータの入力から繰り返して実行する。

【0035】4バイトデータの格納終了時、アドレスが最初のアドレスに対応するものでなく(ステップR5)、最終のアドレスに対応するものでない(ステップR11)ときは、CPU2が格納した4バイトデータからECCデータを生成して(ステップR12)、データバス4を介してEEPROM制御回路7内のECCデータスタックレジスタ72へECCデータを転送し(ステップR13)、EEPROM8内のユーザデータレジスタ81へユーザデータスタックレジスタ71からユーザデータを転送し、ECCデータレジスタ82へECCデータスタックレジスタ72からECCデータを転送し(ステップR14)、EEPROM8への書き込みセット信号を出力して(ステップR15)、EEPROM8のユーザデータ領域83にユーザデータレジスタ81からユーザデータを書き込み、ECCデータ領域84にECCデータ領域82からECCデータの書き込みを行う。そして書き込みが終了した(ステップR16)とき、EEPROM8への書き込みリセット信号を出力し(R17)、ステップR10に戻って、アドレスを+4シフトして(ステップR10)、再びステップR1に戻って、外部からの4バイトデータの入力から繰り返して実行する。ステップR11において、最後のアドレスになったときは、CPU2が格納した4バイトデータからECCデータを生成し(ステップR18)、データバス4を介してEEPROM制御回路7内のECCデータスタックレジスタ72へECCデータを転送し(ステップR19)、EEPROM8内のECCデータレジスタ82へECCデータを転送し(ステップR20)、EEPROM8への書き込みセット信号を出力して(ステップR21)ECCデータの書き込みを行う。そして書き込みが終了した(ステップR22)とき、処理を終了する。

【0036】読み出し時ECCアドレスをシフトする場合の各データの書き込みタイミングは、図15に例示されるようになる。すなわち、ポート1から4バイトからなるページ単位に、ユーザデータを、ページアドレス0, 1, 2, ..., n, ..., 最終アドレスの順にEEPROM8内のユーザデータレジスタ81に転送するとともに、これに基づいて1サイクル遅れて生成された対応するECCデータを、順次EEPROM8のユーザデータレジスタ81に転送し、書き込みパルスに応じて、それぞれEEPROM8の同一アドレスのユーザデータ領域83とECCデータ領域84に書き込む。したがって、この場合は、ページアドレスnのユーザデータに対応するECCデータは、ECCデータ領域84のアドレスn+1に書き込まれる。誤り訂正処理のためにEEPROM8のデータを読み出すときは、EEPROM8のユーザデータ領域83のアドレスnと、ECCデータ領域8

4のアドレスn+1とを読み出すことによって、対応するユーザデータ部分とECCデータ部分とを同時に読み出すことができる。

【0037】読み出し時ECCアドレスをシフトする場合の、EEPROM制御回路7における、EEPROMアドレス生成回路18Aは、図16に示すように、アドレスバス183と、ディクリメント回路184とを含んで構成されている。アドレスバス183は、アドレスバス11の一部であって、アドレス信号を転送する。ディクリメント回路184は、アドレスバス183のアドレスから1減算して出力する。EEPROM8からは、32ビットのユーザデータと、6ビットのECCデータとを同時に読み出す必要があるので、アドレスバス183のビット2~15を、EEPROM8のユーザデータ領域83の読み出しアドレスとして、EEPROM8のアドレス0~13に与え、アドレスバス183のビット2~15を、ディクリメント回路184を介して1を減算して、EEPROM8のECCデータ領域84の読み出し用として、EEPROM8のアドレス0~13に与える。このような変換を行うのは、アドレス信号がバイト単位(8ビット)であるのに対して、EEPROM8の32ビットの入出力データに対応する対応するアドレスとしては、最下位の2ビットが不要なためである。

【0038】次に、図17、図18及び図19を参照して、この例の第4の動作例について説明する。この第4の動作例は、EEPROM8の書き込みが速いため、書き込みパルス幅内でECCデータを生成できない場合に、EEPROM8のECCデータ領域側に、ファームROM5のプログラムに基づくCPU2の処理によって、書き込みアドレスをインクリメントする、書き込みアドレスインクリメントの機能を設ける際の処理方法を示している。ユーザデータを収容した外部のファイル(不図示)における、ユーザデータの最初のアドレスと最後のアドレスとが予め指定されている。最初のアドレスから入出力ポート1へユーザデータを入力して(ステップQ1)、データバス4を介してEEPROM制御回路7内のユーザデータスタックレジスタ71へユーザデータを転送するとともに(ステップQ2)、入出力ポート1からのユーザデータをデータバス4を介してCPU2の内部レジスタ群21へ格納する(ステップQ3)処理を繰り返すことによって、4バイトデータを格納した(ステップQ4)後、ユーザデータスタックレジスタ71から4バイトデータをEEPROM8内のユーザデータレジスタ81へ転送する(ステップQ5)。

【0039】一方、ステップQ5で最初の4バイトデータをEEPROM8内のユーザデータレジスタ81へ転送したとき、内部レジスタ群21からデータバス4を介してユーザデータスタックレジスタ71に対応するアドレスのRAM6の領域へ4バイトデータを格納する(ステップQ6)。そして、データバス4を介してユーザデ

ータスタックアドレスが示すRAM6の領域から4バイトデータを読み出してCPU2へ転送する(ステップQ7)。これによってCPU2は、4バイトデータからECCデータを生成して(ステップQ8)、データバス4を介してEEPROM制御回路7内の対応するアドレスのECCデータスタックレジスタ72へECCデータを転送し(ステップQ9)、さらにEEPROM8内のECCデータレジスタ82へECCデータを転送して、スタックアドレスをデクリメントする(ステップQ11)処理を繰り返して行う。そして、EEPROM制御回路7からEEPROM8のユーザデータ領域83とECCデータ領域84のアドレスを出力し(ステップQ12)、EEPROM8への書き込みセット信号を出力して(ステップQ13)書き込みを行い、書き込み完了(ステップQ14)時、書き込みリセット信号を出力して(ステップQ15)書き込みを終了する処理を、アドレスを4インクリメントし(ステップQ17)ながら繰り返し実行する。ユーザデータが最終アドレスになった(ステップQ16)とき、ECCデータスタックレジスタ72からEEPROM8内のECCデータレジスタ82へECCデータを転送して、転送終了した(ステップQ18)とき、書き込みセット信号を出力して(ステップQ19)EEPROM8のECCデータ領域84に対する書き込みを行って、書き込みが完了した(ステップQ20)とき、書き込みリセット信号を出力する(ステップQ21)処理を、ECCデータ領域84の最終アドレスになる(ステップQ22)まで行って、処理を終了する。

【0040】第4の動作例の場合の各データの書き込みタイミングは、図19に例示されるようになる。すなわち、ポート1からユーザデータを、4バイトからなるページ単位に、ページアドレス0, 1, 2, ..., n, ..., 最終アドレスの順にEEPROM8内のユーザデータレジスタ81に転送するとともに、これに基づいて生成されたECCデータを、生成された時点で順次EEPROM8のECCデータレジスタ82に転送する。ユーザデータは書き込みパルスに応じてそのままEEPROM8のユーザデータ領域83に書き込まれるが、ECCデータに対しては、最初に生成されたECCアドレス0への書き込み(1), (2)は、ECCデータ生成の遅れに基づくデフォルト値なので書き込みは行われず、次のECCアドレス0への書き込み以降のデータが、書き込みパルスに応じて、EEPROM8の所定のアドレスのECCデータ領域84に書き込まれる。

【0041】この動作例の場合は、CPU2によるECCデータの生成が、次のECCデータの書き込みのためのアドレスのインクリメントまでに間に合わなかったときは、CPU2はデフォルト値を出力し、EEPROM制御回路7は、この場合は、EEPROM8のECC領域84への書き込みを行わないようにする。デフォ

ルト値としては、例えば最初のECCデータが生成されるまではオール1を用い、以後は、前回のECCデータをそのままデフォルト値として出力する。誤り訂正処理のためにEEPROM8のデータを読み出すときは、
05 同一アドレスから、ユーザデータと、ECCデータとを読み出すことによって、対応するユーザデータ部分とECCデータ部分とを読み出すことができる。

【0042】このようにこの例によれば、ECCデータを作成する際に、外部のソフトウェアによる処理を行うのではなく、1チップマイクロコンピュータの内部のソフトウェアによって処理を行うので、外部からECCデータを書き込む必要がなく、外部からの書き込みデータ量が少なくなり、したがって、書き込みデータを転送する際のエラーが減少し、データの信頼性を向上することができる。さらにこの場合、EEPROMに対して外部から書き込むデータ量が減少することによって、データの書き込み時間が減少し、生産性が向上する。EEPROMにデータ書き込む外部装置の実装メモリは、ECCデータの分を必要とせず、書き込もうとするユーザデータの分だけあればよいので、この外部装置のコストの負担が少ない。

【0043】以上、この発明の実施例を図面により詳述してきたが、具体的な構成はこの実施例に限られたものではなく、この発明の要旨を逸脱しない範囲の設計の変更等があってもこの発明に含まれる。例えば、この発明の適用は、1チップマイクロコンピュータに限られるものでなく、EEPROMを内蔵してプログラムを外部的に格納するコンピュータに対して、一般的に応用可能なものである。

【0044】

【発明の効果】以上説明したように、この発明の構成によれば、EEPROMにECCデータを付加したデータからなるプログラムを格納して、プログラム読み出し時誤り訂正を行う1チップマイクロコンピュータ等において、1チップマイクロコンピュータ等の内部でECCデータを生成するようにしたので、外部的にECCデータを生成して付加する場合と比べて、ユーザの負担が大幅に減少する。また、ECCデータ作成のために、作成したECCデータを一旦格納する外部メモリを必要としないので、システムの装置のコストを低減できる。さらに、EEPROMに対して、ユーザデータとECCデータとを同時に書き込むので、ECCデータを外部メモリから書き込む場合と比べて、EEPROMに対するデータ書き込み時間を低減することができる。

【0045】また、従来のようにECCデータの生成を論理回路で構成したECCデータ生成回路で行う場合と異なり、ECCデータの生成用のファームROMを追加するだけでよいので、回路規模が大幅に増加することがなく、1チップマイクロコンピュータのチップサイズを増大させることもない。また、ECCデータの生成処理

は、プログラム格納時にのみ動作し、通常のプログラム実行時には、ECCデータの生成処理のファームROMは動作しない。また、プログラム実行時には、論理回路で構成した誤り訂正回路を用いて誤りの検出と訂正の処理を行うので、通常のプログラム処理速度を低下させることはない。

【0046】また、通常、EEPROMにデータを書き込む場合、RAMに比べて数100倍～数1000倍の書き込み時間を必要とする。この発明では、外部から受け取るユーザデータを複数バイトまとめてEEPROMに書き込んだり、複数バイトのユーザデータとECCデータとをまとめてEEPROMに書き込むようにしたので、ユーザデータを高速に受信でき、また書き込み時間を利用してECCデータを計算することが同時並行的に処理可能になる。

【図面の簡単な説明】

【図1】この発明の一実施例である1チップマイクロコンピュータの電気的構成を示す図である。

【図2】EEPROM制御回路の構成例を示す図である。

【図3】データラッチ選択制御回路の構成例を示す図である。

【図4】書き込み信号生成回路の構成例を示す図である。

【図5】書き込み信号生成回路の動作タイミングを示す図である。

【図6】この例の第1の動作処理手順を示すフローチャートである。

【図7】この例の第1の動作処理手順を示すフローチャートである。

【図8】第1の動作例における書き込み時のタイミングチャートを示す図である。

【図9】この例の第2の動作処理手順を示すフローチャートである。

【図10】この例の第2の動作処理手順を示すフローチャートである。

【図11】書き込み時ECCアドレスをシフトする場合のタイミングチャートを示す図である。

【図12】書き込み時ECCアドレスをシフトする場合のEEPROMアドレス生成回路の構成例を示す図である。

【図13】この例の第3の動作例のフローチャートを示す図である。

【図14】この例の第3の動作例のフローチャートを示す図(2)である。

【図15】読み出し時ECCアドレスをシフトする場合のタイミングチャートを示す図である。

【図16】読み出し時ECCアドレスをシフトする場合のEEPROMアドレス生成回路の構成例を示す図である。

【図17】この例の第4の動作処理手順を示すフローチャートである。

【図18】この例の第4の動作処理手順を示すフローチャートである。

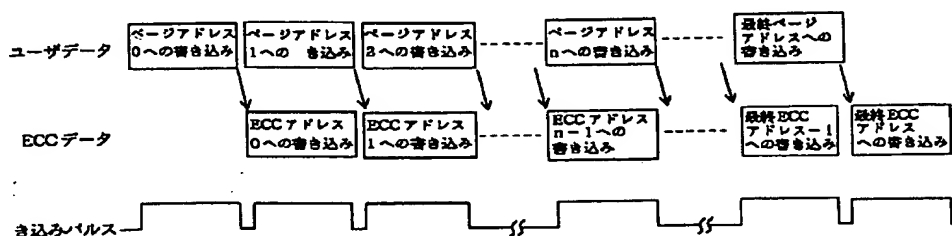
【図19】第4の動作例における書き込み時のタイミングチャートを示す図である。

【図20】EEPROMにおける従来のECCデータの生成・付加方法を説明するための説明図である。

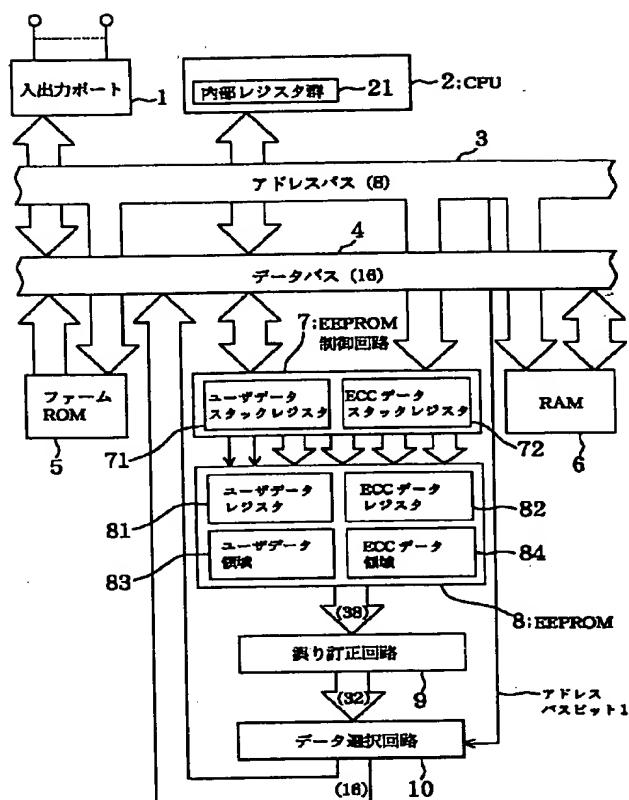
【符号の説明】

- 1 入出力ポート
- 2 CPU
- 2 1 内部レジスタ群
- 3 アドレスバス
- 4 データバス
- 5 ファームROM (記憶手段)
- 6 RAM
- 7 EEPROM制御回路 (制御手段)
- 30 7 1 ユーザデータスタックレジスタ
- 7 2 ECCデータスタックレジスタ
- 8 EEPROM
- 8 1 ユーザデータレジスタ
- 8 2 ECCデータレジスタ
- 35 8 3 ユーザデータ領域
- 8 4 ECCデータ領域
- 9 誤り訂正回路
- 10 データ選択回路

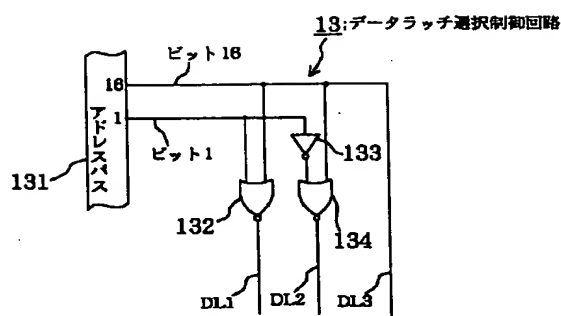
【図11】



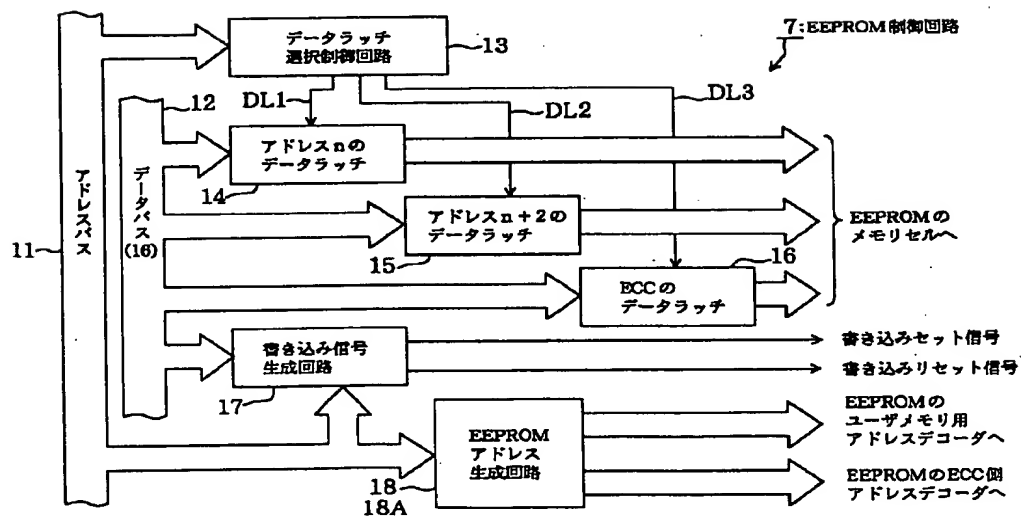
【图 1】



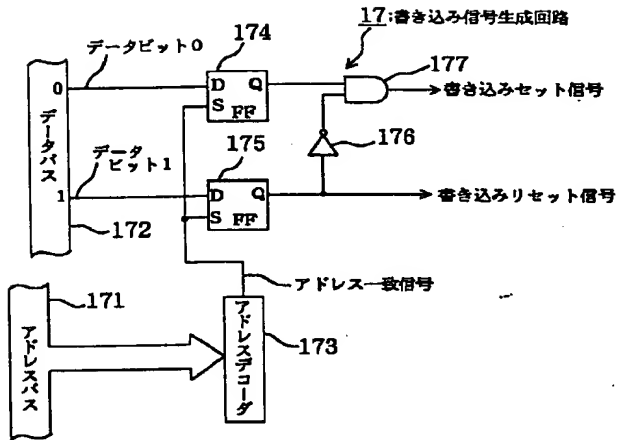
【図 3】



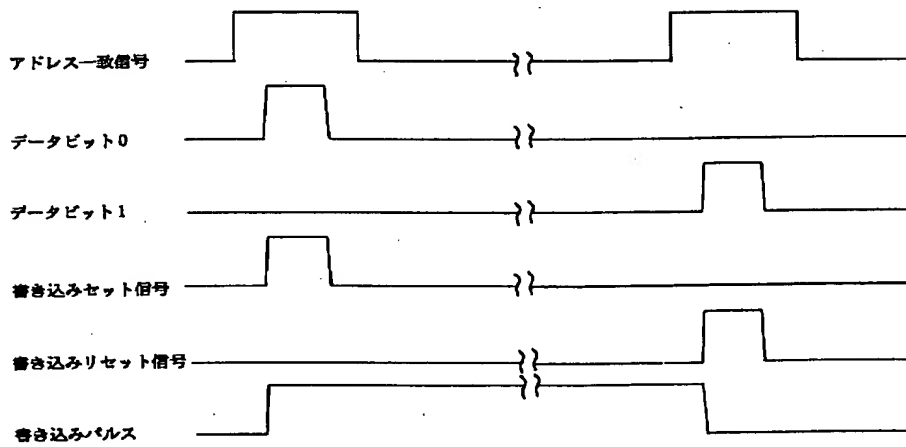
【図 2】



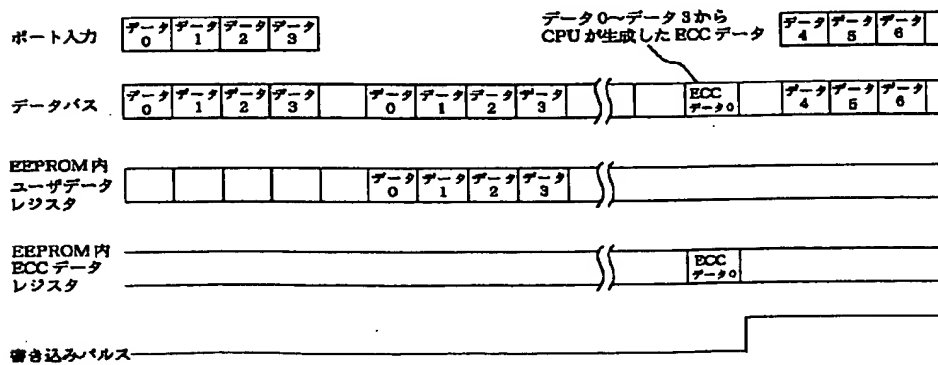
【図 4】



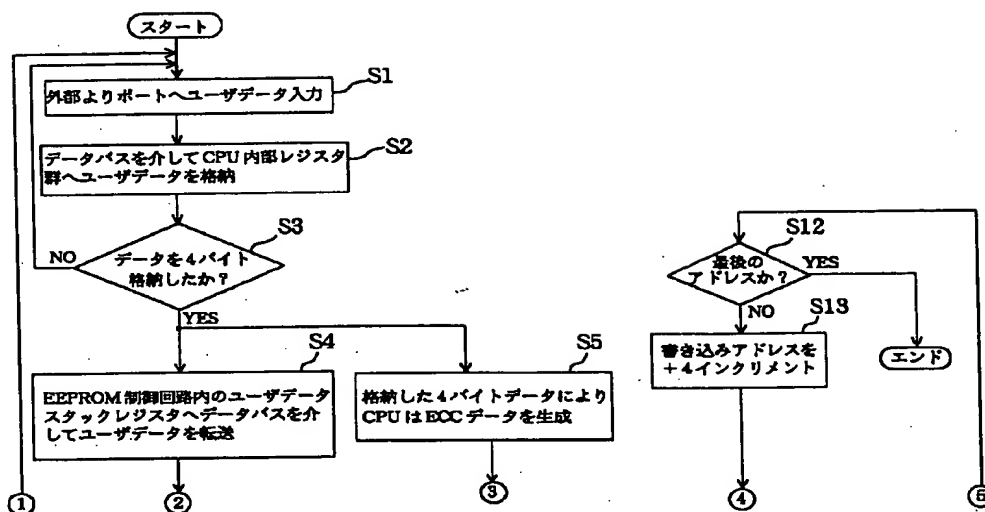
【図 5】



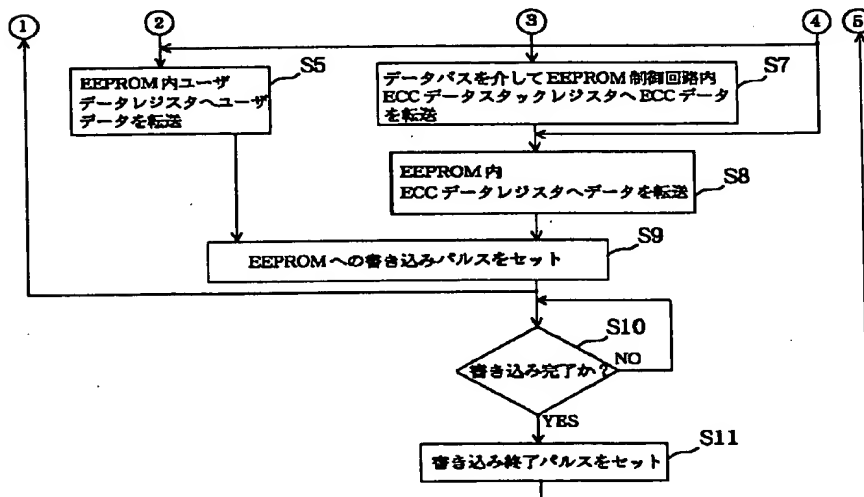
【図 8】



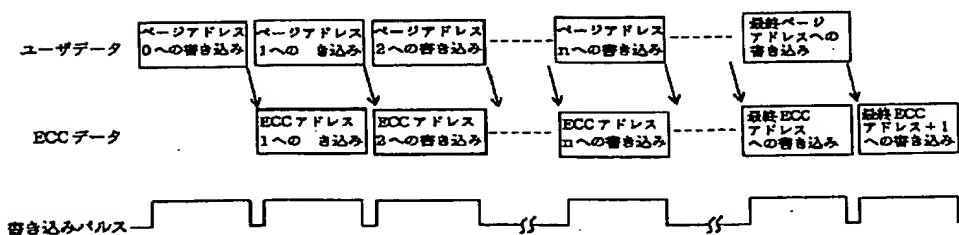
【図6】



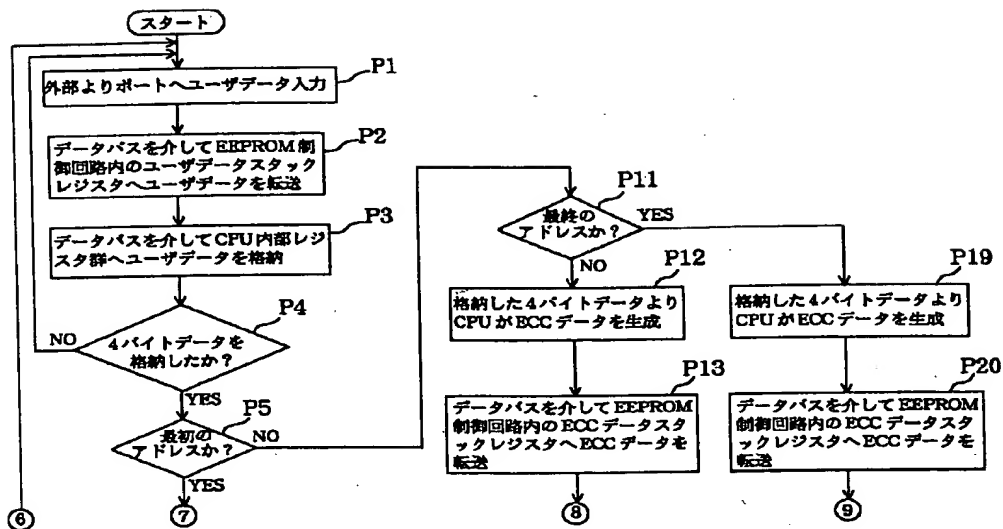
【図7】



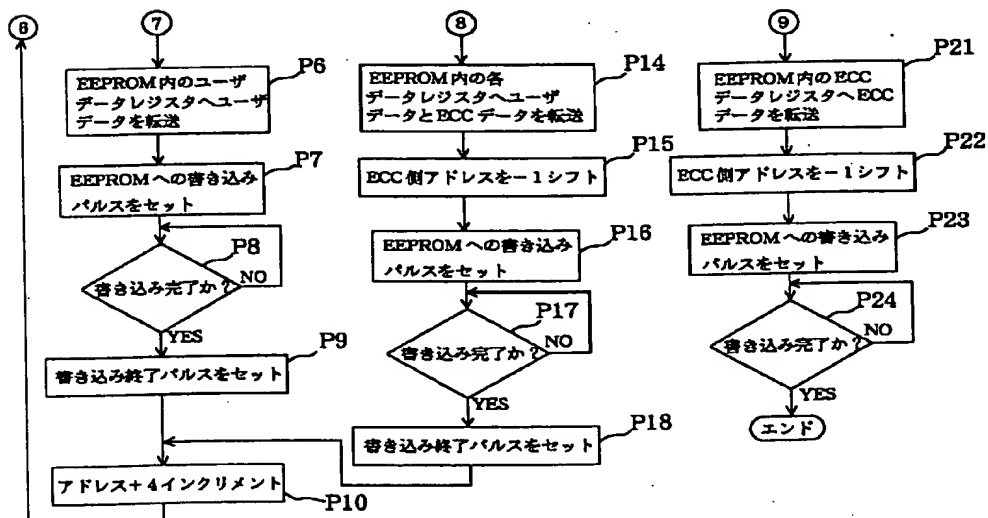
【図15】



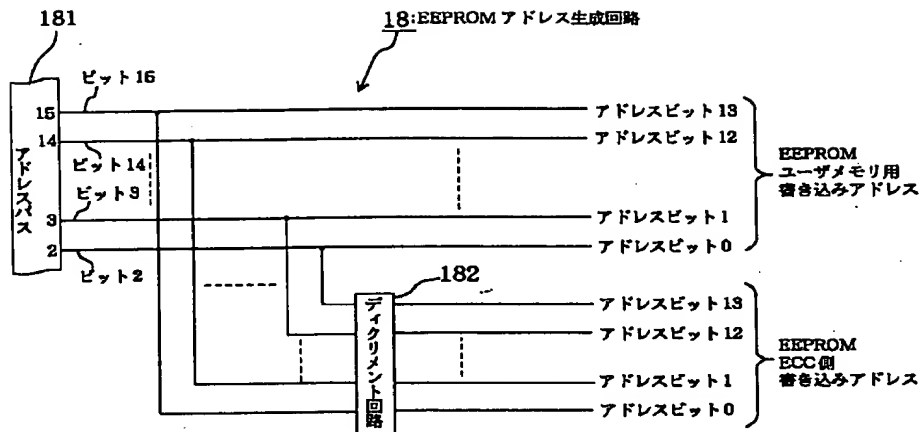
【図9】



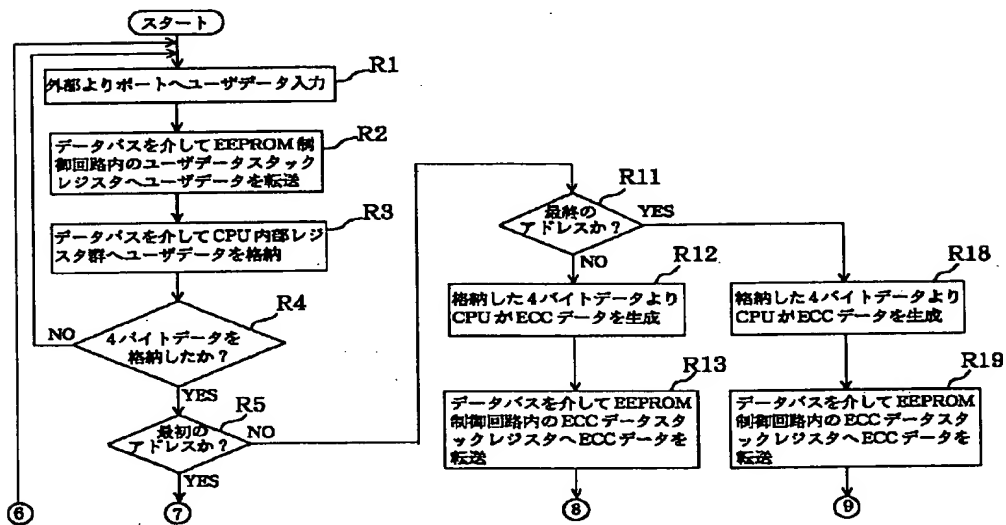
【図10】



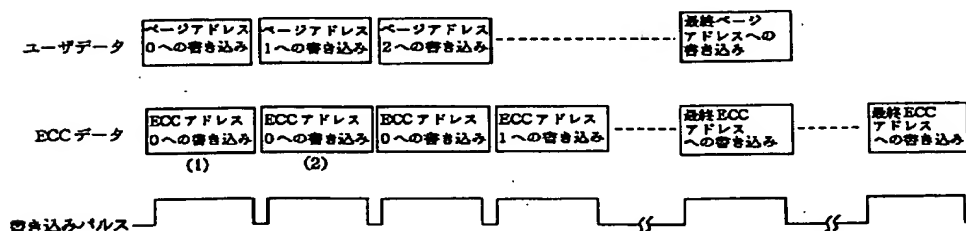
【図12】



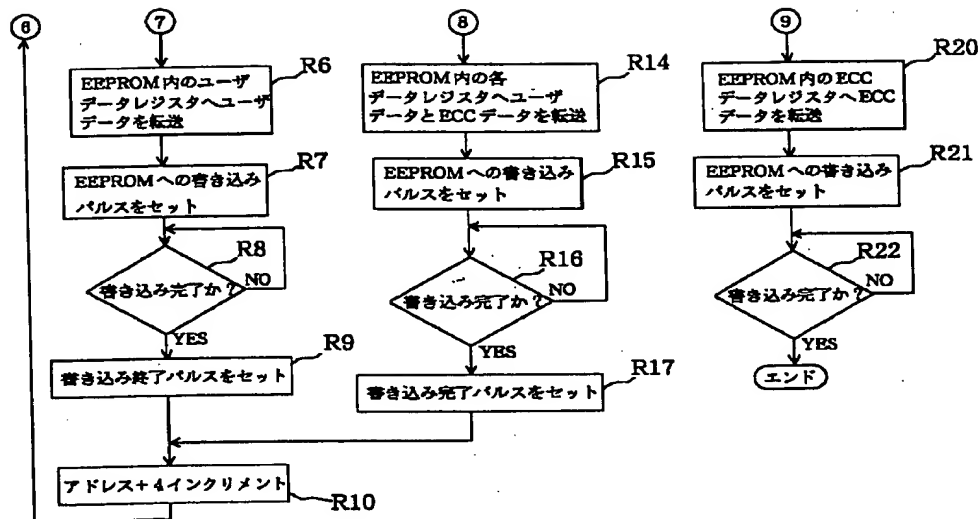
【図13】



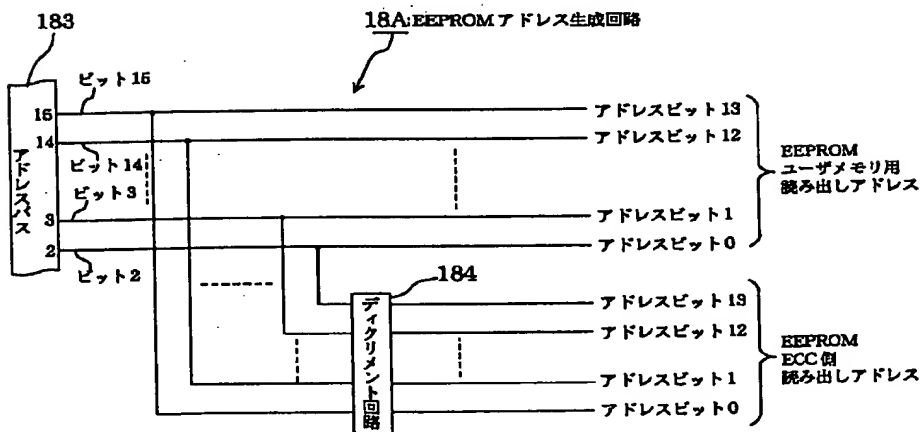
【図19】



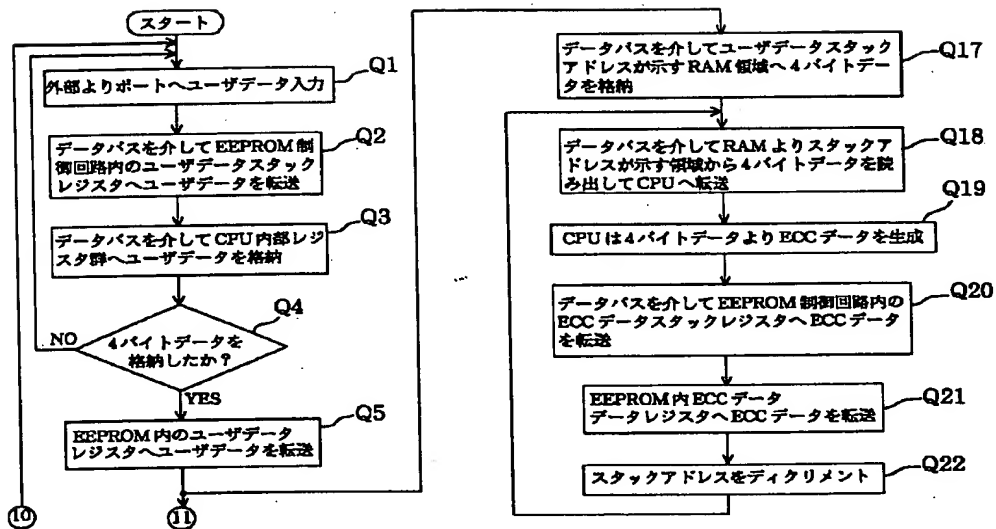
【図14】



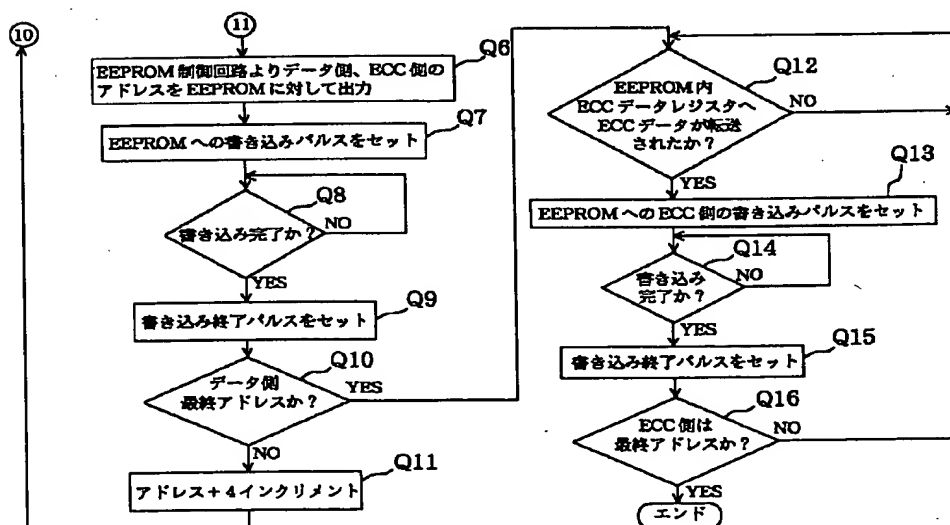
【図16】



【図17】



【図18】



【図 2 0】

